



## Course Title:

### **MATLAB Programming Techniques**

## Course Purpose:

This two-day course provides hands-on experience using the features in the MATLAB® language to write efficient, robust, and well-organized code. These concepts form the foundation for writing full applications, developing algorithms, and extending built-in MATLAB capabilities. Details of performance optimization, as well as tools for writing, debugging, and profiling code are covered. Topics include:

- Structuring data
- Managing data efficiently
- Utilizing development tools
- Structuring code
- Creating robust applications
- Verifying application behavior

## Pre- requisites:

MATLAB Fundamentals



- ✓ 2 training days
- ✓ Hours: 09:00-17:00
- ✓ Total training hours: 16

## Teaching method

The course combines lectures, demonstrations and practical exercises in MATLAB, using original training books from MathWorks. The course is in Hebrew but the training materials are in English.

עמוד מס' 1

### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 6 **Email:** [training@systematics.co.il](mailto:training@systematics.co.il)

**Website:** <http://www.systematics.co.il/mathworks>



## **Course Objective:**

### **Structuring Data**

**Objective:** Explore choices for storing data within a MATLAB application.

- Data types
- Concatenation and conversion
- Cell and structure arrays
- Extracting data

### **Managing Data Efficiently**

**Objective:** Use common techniques for improving performance when storing, accessing, and processing data.

- Preallocation
- Vectorization
- Memory management
- Objects and classes

### **Utilizing Development Tools**

**Objective:** Use integrated tools to manage code, diagnose problems, debug applications, and measure performance.

- Managing file with Projects
- Finding potential problems in code
- Debugging code
- Assessing code performance
- Using Source Control

### **Creating Robust Applications**

**Objective:** Create robust applications that withstand unexpected input and produce meaningful errors. Use built-in MATLAB functions and programming constructs, and employ standard techniques for handling error conditions.

- Creating flexible function interfaces
- Checking for warning and error conditions
- Working with the try-catch construct and MException objects

### **Structuring Code**

**Objective:** Modularize code into readable and maintainable functions. Explore the different function types available in the MATLAB language and the tradeoffs

עמוד מס' 2

### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 6 **Email:** [training@systematics.co.il](mailto:training@systematics.co.il)

**Website:** <http://www.systematics.co.il/mathworks>



associated with each type. See how function type affects the performance and reliability of the code.

- Private functions
- Local functions
- Nested functions
- Function handles
- Precedence rules
- Comparison of function types

### Verifying Application Behavior

**Objective:** Create tests to verify that code meets requirements.

- Writing test scripts
- Running test scripts
- Using test functions

### Appendix A: Creating a Toolbox

**Objective:** Package code and documentation into a custom toolbox for distribution to other users.

- Creating documentation
- Packaging the toolbox
- Distributing the toolbox

עמוד מס' 3

#### **Training Center Systematics - Contact information:**

**Phone number:** 03-7660111 Ext: 6 **Email:** [training@systematics.co.il](mailto:training@systematics.co.il)

**Website:** <http://www.systematics.co.il/mathworks>